Flappy Bird és un joc de mòbil que es va popularitzar l'any 2013 que consisteix en controlar un ocell que ha de volar entre una sèrie de tubs que apareixen a la pantalla. Per mantenir-lo en vol, el jugador ha de tocar la pantalla enlairant l'ocell lleugerament, si no es toca, l'ocell cau a causa de la gravetat. L'objectiu principal és passar entre els tubs sense xocar-hi ni caure al terra i cada cop que l'ocell passa amb èxit entre dos tubs, el jugador guanya un punt.

Petita introducció als inputs

Quan la xarxa neuronal només coneix la posició Y de l'ocell (input 0), els resultats són completament aleatoris anant desde 0 fins a 4000 de fitness. Mai s'apropa a una dada superior 4000 de fitness, ja que no té la informació necessària per arribar al seu objectiu, degut a que amb les dades que posseeix no pot saber en quina posició es troba el forat de l'obstacle i l’unic que pot fer es evitar caure al terra i estar a una altura en la qual pota haver un forat.

En aquest cas de només donar la posició Y de l’obsacle (Input 1), la xarxa neuronal sap a quina alçada està l'obstacle, però no la posició de l'ocell, per la qual cosa no és capaç de superar-lo ni de manera aleatòria, com sí que passava amb l'input 0.

Quan la xarxa neuronal coneix la posició Y de l'ocell i la posició Y de l'obstacle (input 0-1), l'aprenentatge de la xarxa és més ràpid i efectiu, ja que té la informació necessària per completar la tasca. D’altre banda, un sol input que relaciona aquestes dos variables de forma directa, és més eficient, millora la velocitat d’aprenentatge i redueix la complexitat de la xarxa neuronal oferint uns millors resultats acord amb les gràfiques. (sha de fer gràfiques)

Quan oferim la posició Y de l'ocell i la posició Y i X de l'obstacle (Input 0-1-2) el temps d'aprenentatge augmenta respecta amb la configuració d'inputs 0-1 degut a l’augment de complexitat de la xarxa neuronal

Si combinem la posició Y de l'ocell, la posició Y de l'obstacle i la velocitat Y de l'ocell (Input 0-1-3) passa a ser una configuració menys efectiva que la anterior, ja que l'aprenentatge es ralentitza més per la presència de la velocitat Y, un input que la xarxa tendeix a ignorar o eliminar.

Quan ajuntem ja tots els inputs posició Y de l'ocell, posició Y i X de l'obstacle, velocitat Y de l'ocell (input 0-1-2-3) s’alenteix l’aprenentatge i passa a ser una de les xarxes neuronals menys optimes, traient de banda les que no tenen prou informació, degut a que ha de ignorar el excés d’informació.

Parlar de la senzillesa de la xarxa neuronal, lineal ... com mes complexitat mes lent aprenentatge.

Introducció mutacions (crec que ja la ha fet el marc)

Mutació no estructural (Mutació 0):

Mutació estructural (Mutació 1):

Mutació crossover (Mutació 2):

La aplicació de la mutació no estructural (Mutació 0) en general provoca unes puntuacions de fitness més altes i uns valors més sòlids/estables que varien poc.

Quan apliquem la mutació estructural (Mutació 1) provoca uns valors més dispersos i menys sòlids/estables i, en conseqüència unes puntuacions de fitness més baixes.

D’altre banda, la implicació de la mutació crosteres (Mutació 2) no provoca canvis notables en la majoria de partides; tot i això, en un 20% dels casos on la mutació 0 i la mutació 2 es troben en les mateixes partides la puntuació es dispersa exageradament sense cap motiu aparent.

Mirant tota la sèrie de gràfics es poden observar diferents tipus de comportament a l'hora d'implicar els diferents tipus de població, les diferents mutacions que hi ha són:

Com més població hi ha en una generació més ràpida és la capacitat que per la IA d'aprendre, a causa de la presència de més xarxes neuronals que permetien que aprengués més de pressa.